

# 効果的な分散型データベース設計について

成 久 洋 之\* 青 江 俊 夫\*\*

\*岡山理科大学 電子工学科

\*\*順正短期大学 幼児教育科

(昭和62年 9 月30日 受理)

## 1. はじめに

情報処理分野におけるハードウェアおよびソフトウェアの発展は、ユーザに対して情報（データ）の収集法と利用技術に多大な貢献をしてきた。我々の周辺において、コンピュータの利用環境が数年前に比較して格段に整備されるにともない、さまざまな種類の情報が大量に発生し、収集されるようになってきた。これら各種の情報をどのように収集し、格納すれば良いのであろうか？この疑問を解決するべくデータベースの概念が導入された。

当初データベースは、中央集中型、すなわち一カ所に情報を集約し、そのシステムの管理者の管理の下で他のユーザがデータを利用させてもらうという形態であった。しかし現在では、パソコンを含めて各種のコンピュータが手軽に利用できるようになり、各拠点からの効率的な情報のアクセスという観点から分散型のデータベースの概念が発達した。

分散型データベースの概念は、“コンピュータネットワーク上に存在するいくつかの拠点に分散配置されたデータベースを、複数の拠点間で相互に参照したり利用できるように設計されたデータベースである”と定義できる。その時、個々の利用者にとっては、分散化されたデータベースのすべてがあたかも利用者プログラムが動作している拠点に仮想的に存在するかのように見えるシステム形態である。

データベースを分散型として設計することにより、従来の中央集中型と比較して、データベースの構築時、データの参照時点あるいは特定の情報のアップデート時点において、情報の処理時間や情報の蓄積または修正のために情報を転送する際の経費が大幅に削減できるものと期待される。

本研究では、効果的な分散型データベースの設計にあたって分散化の指針となる情報の收拾について、コストの定式化とそれに基づくシミュレーション実験を行ったので報

告する。

## 2. 問題の定式化

データベースに収容したファイルをいくつかの拠点に分散させて情報の相互参照を行う場合、当然、各拠点にはデータベースを処理できる能力を持ったコンピュータ・システムが設置されているものと仮定する。そしてそれらの処理設備は通信回線で結合されてコンピュータ・ネットワークを構成し、ネットワーク上の他の拠点に存在するデータベースの情報をオンラインで自由にアクセスできるものとする。

通信回線によるネットワークとしては、一般に NTT の専用回線や電話回線を利用する場合が考えられるが、この場合情報の送受信に必要な回線のコストとしては、通常、回線の結合回数あるいは 1 回の通信量や結合時間によって変動する部分と、毎月固定的に付加される基本料金のような定額部分がある。定額部分については、回線の利用状況によって 1 回当りの相対コストに大きく影響を与えるので、シミュレーション実験には不適当な要因となるので本研究では削除して考えた。また、データは 1 回の送受信において 128 オクテットを転送すると仮定してモデル化をおこなった。

データベースを分散化したときの効率を測定する尺度として何を使用するかという問題がある。例えば、1 回の処理要求当りについて考えるか全処理要求を対象として考えるか、あるいは処理時間を見るのか処理に関わるコストを見るのか等いろいろ考えられるが、本研究では処理コストを中心に考えることにする。

データベースを使用して、与えられた問題を解決する際に必要と考えられるコストとしては、

- (1) データベースの格納コスト
- (2) データベースを読む込むコスト
- (3) データベースを変更するコスト
- (4) データベースの追加・削除コスト

等が考えられる。以下でこれらのコストの定義式について考察をする。

### 2. 1 格納コスト

データベースは、不特定多数のユーザーが、希望するときに自由に利用できなければ意味がない。そのために、データベースに関するすべての情報を補助記憶装置（通常、磁気ディスク装置）上に常駐させておく必要がある。それ故、データベースが存在するために各拠点でのユーザ領域は制限を受けることになる。そこで補助記憶装置の占有量にしたがって何等かの使用料金をペナルティとして科せる必要がある。本研究では、実

使用論理ブロック数を基準にして算定することにした。

全拠点を通して、全てのファイルを格納するために必要なコストは次式で得られる。

$$C_1 = \sum_{k=1}^P \sum_{j=1}^S b_{nkj} \cdot C_{1kj} \quad (2.1)$$

ここで  $b_{nkj}$  は、拠点  $k$  においてファイル  $j$  を格納するために必要なブロック数であり、

変数名	定 義
A	属性数
R	レコード数
S	分割されたファイルの数
T	処理リクエストの数
v	ベリファイをするときは1 その他は0
P	拠点数
Z	同一拠点間の転送なら1、その他は0
$X_{ijk}$	ファイル $j$ に属性が存在するとき1 その他は0
$b_{rki}$	ファイル $j$ のブロッキング・ファクタ
$b_{lki}$	ファイル $j$ のブロック長
$b_{nkj}$	ファイル $j$ の占めるブロック数
$C_1$	格納コスト
$C_{1kj}$	ファイル $j$ の1ブロックの格納コスト
$C_2$	読み込みコスト
$C_{2kj}$	読み込み処理のバイト当りの転送コスト
$C_{2stkj}$	// ヘッド移動コスト
$C_{2wtkj}$	// ブロック転送コスト
$C_{2ftkj}$	// 固定経費
$C_3$	変更処理コスト
$C_{3kj}$	変更処理のバイト当りの転送コスト
$C_{3stkj}$	// ヘッド移動コスト
$C_{3wtkj}$	// ブロック転送コスト
$C_{3ftkj}$	// 固定経費
$C_4$	追加・削除コスト
$C_{4kj}$	追加・削除処理のバイト当りの転送コスト
$C_{4stkj}$	// ヘッド移動コスト
$C_{4wtkj}$	// ブロック転送コスト
$C_{4ftkj}$	// 固定経費
$h_{mki}$	情報を読みながらヘッドを移動するコスト
$h_{oki}$	情報をパスしてヘッドを移動するコスト
$m_{eki}$	ファイル $j$ がある装置の記憶容量
$n_{kj}$	ファイル $j$ の格納に必用な記憶容量
$p_{tki}$	シリンダ内の1つのレコードが処理要求を満足する確率
$q_{tki}$	どのレコードも処理要求を満足しない確率
$r_{eki}$	1シリンダに収容できるレコード数
$y_{kj}$	データ送受に関わる潜在的なコスト
$u_{ikt}$	属性が修正されたとき1、その他は0
$t_{ck}$	処理要求を出した拠点と、拠点 $k$ 間の転送経費
$\mu_n$	各処理コストを調整するための重み

表1 変数表

$c_{lkj}$ は拠点  $k$  においてファイル  $j$  の 1 ブロックを格納するために必要なコストを示す。  
 なお、本論文中の数式に現れる各種の変数の定義内容は表 1 に概説する。

## 2. 2 データベース読み込みコスト

この項で扱う“読み込み”とは、集計の目的、あるいは一覧表の作成など、受け取った情報に対しては一切の変更を伴わない場合を指している。

読み込みコストは、磁気ディスクの Read/Write ヘッドの移動コスト、メモリ上に 1 論理ブロックを転送するコスト、ファイルのアクセスに関わる基本的なコストの 3 種類に分類する。

### (1) ヘッド移動コスト

拠点  $k$  のファイル  $j$  内のあるシリンダ中の、少なくともいずれか 1 個のレコードが現在発行されている処理要求を満足する場合の確率を  $p_{tkj}$  とすると、逆に拠点  $k$  のファイル  $j$  のシリンダ中のいずれのレコードも処理要求を満足しない場合の確率は、 $(1 - p_{tkj})$  で表すことが出来る。

拠点  $k$  のファイル  $j$  において、1 シリンダ内に含まれるレコード数を  $r_{ckj}$  とすると、ファイル  $j$  の全レコードを格納するために必要なシリンダ数は、 $[R/r_{ckj}]$  となる。ここで記号  $[ \cdot ]$  はガウスの記号を意味し、カッコ内の式の値に等しいかそれより大きい最小の整数値を表している。

したがって、ヘッドを移動するためのコストは、

$$C_{2stkj} = \left[ \frac{R}{r_{ckj}} \right] \cdot p_{tkj} \cdot h_{mkj} + \left[ \frac{R}{r_{ckj}} \right] \cdot (1 - p_{tkj}) \cdot h_{pkj} \quad (2.2)$$

で定義される。

### (2) 読み込みコスト

いま、処理リクエストが同一拠点内のファイルを対象とするとき値 0 を与え、他の拠点のファイルを対象とする場合に 1 を与えるような  $1 \cdot 0$  の変数  $z$  を導入すると、同一拠点内での読み込みコストは、

$$(1 - z) \cdot b_{lkj} \cdot c_{2kj} \quad (2.3)$$

、他の拠点から情報を読み込む場合のコストは、

$$z \cdot \left[ \frac{b_{lkj}}{128} \right] \cdot c_{2kj} \quad (2.4)$$

としてそれぞれ別個に計算できる。両式より、1 ブロックを読み込むコストは、

$$t_{ck} = c_{2kj} \cdot \left\{ (1 - z) \cdot b_{lkj} + z \cdot \left[ \frac{b_{lkj}}{128} \right] \right\} \quad (2.5)$$

となる。

したがって、現在のリクエストで要求されているファイルの全ブロックを読み込むコストは、

$$C_{2wtkj} = \left[ \frac{R}{b_{fkj}} \right] \cdot (1 - (q_{tkj})^{b_{fkj}}) \cdot (y_{kj} + t_{ck}) \quad (2.6)$$

で計算される。

### (3)基本経費

データベースをアクセスするために当然ファイルを使用できる状態にしておかなければならないし（いわゆる Open 処理）、また使用後は他の利用者が利用できるようにファイルを解放しなければならない（Close 処理）。ここで扱う基本経費とは、ファイルの Open/Close 処理に伴う経費を表し、固定的な費用と考えられるので、

$$C_{2ftkj} \quad (2.7)$$

と定義する。

したがって、データベースを読み込むコストは次式で与えられる。

$$C_2 = \sum_{t=1}^T \sum_{k=1}^P \sum_{j=1}^S (C_{2stkj} + C_{2wtkj} + C_{2ftkj}) \quad (2.8)$$

## 2.3 変更コスト

データベースの変更コストは、読み込みの場合と基本的には同一の考え方を適用するが、読み込んだレコード中のいずれかの属性をアップデートした後、もう一度同じ場所に書き戻すと共に、必要なら正しく修正されたかどうかを検証する点が異なる。

### (1)ヘッド移動コスト

読み込みの場合と同じで、

$$C_{3stkj} = \left[ \frac{R}{r_{ckj}} \right] \cdot p_{tkj} \cdot h_{mkj} + \left[ \frac{R}{r_{ckj}} \right] \cdot (1 - p_{tkj}) \cdot h_{pkj} \quad (2.9)$$

となる。

### (2)修正コスト

レコードの修正作業は、該当レコードをメモリに読み込み、変更処理を受けた後で元の位置に書き戻され、必要ならば検証を受ける。したがって、修正コストはこれら3つの処理コストの合計で求められる。

#### ○読み込みコスト

(2.6)の読み込みコストと同じ、

$$C_{3wtkj} = \left[ \frac{R}{b_{fkj}} \right] \cdot (1 - (q_{tkj})^{b_{fkj}}) \cdot (y_{kj} + t_{ck}) \quad (2.10)$$

で与えられる。

○書き戻しコスト

修正したレコードを含むブロックを元の位置に書き戻すコストは,

$$\max_i(u_{ikt} \cdot x_{ikj}) \cdot (y_{kj} + t_{ck} + v \cdot t_{ck}) \quad (2.11)$$

となる。従ってデータの修正コストは,

$$\begin{aligned} C_{3wtkj} = & \left[ \frac{R}{b_{f kj}} \right] \cdot \{(1 - (q_{tkj})^{b_{f kj}}) \cdot (y_{kj} + t_{ck}) \\ & + \max_i(u_{ikt} \cdot x_{ijk}) \cdot (y_{kj} + t_{ck} + v \cdot t_{ck})\} \end{aligned} \quad (2.12)$$

となる。

(3)基本経費

ファイルのアクセスに関する経費は読み込みの場合と同じで,

$$C_{3ftkj} \quad (2.13)$$

である。

これらを総合して、データベースを修正する際に必要な経費は,

$$C_3 = \sum_{t=1}^T \sum_{k=1}^P \sum_{j=1}^S (C_{3stkj} + C_{3wtkj} + C_{3ftkj}) \quad (2.14)$$

で与えられる。

## 2. 4 追加・削除コスト

追加・削除処理は、該当ファイルをアップデートしながら全レコードを新しいファイルに複写するものとする。

(1)ヘッド移動コスト

新・旧ファイルにおいて、ヘッドの移動はファイル内の全ブロックに渡って位置づけを行わなければならないから、そのコストは,

$$C_{4stkj} = 2 \cdot \left[ \frac{R}{r_{ckj}} \right] \cdot h_{mkj} \quad (2.15)$$

となる。

(2)データ複写コスト

旧ファイルから新ファイルにデータを複写するためのコストは、前出の(2.12)式において全レコードを対象とした場合に当たるので、第2項の確率が1の時である。故に,

$$C_{4wtkj} = \left[ \frac{R}{b_{fjkj}} \right] \cdot \{(y_{kj} + t_{ck}) + \max_i (u_{ikt} \cdot x_{ijk}) \cdot (y_{kj} + t_{ck} + v \cdot t_{ck})\} \quad (2.16)$$

である。

### (3)基本経費

新・旧ファイルをアクセスするから、経費は、

$$C_{4ftkj} \quad (2.17)$$

となる。

したがって、データベースへの追加・削除に関するコストは、

$$C_4 = \sum_{t=1}^T \sum_{k=1}^P \sum_{j=1}^S (C_{4stkj} + C_{4wtkj} + C_{4ftkj}) \quad (2.18)$$

で与えられる。

## 2.5 付加条件

本研究においては、次のような制約条件を導入する。

(1)ある1つの属性は、分散したいずれかの拠点のデータベースに関するファイルのどこか1カ所にのみ存在する。したがって、

$$x_{fjk} = \begin{cases} 1: \text{属性 } i \text{ が拠点 } k \text{ のファイル } j \text{ 上に存在する場合} \\ 0: \text{そうでない場合} \end{cases} \quad (2.19)$$

$$(i=1, 2, \dots, A; \quad j=1, 2, \dots, S; \quad k=1, 2, \dots, P)$$

と定義すると、データベース全体を通して考えると、

$$\sum_{k=1}^P \sum_{j=1}^S x_{ijk} = 1 \quad (i=1, 2, \dots, A) \quad (2.20)$$

である。

(2)各拠点でのデータベース格納のための補助記憶装置の記憶容量に関する制限から、

$$n_{kj} \leq m_{ckj} \quad (j=1, 2, \dots, S; \quad k=1, 2, \dots, P) \quad (2.21)$$

すなわち、拠点  $k$  のファイル  $j$  が占有できる記憶容量は  $j$  が割り当てられている装置の記憶容量あるいは使用許可容量を越えてはならないことである。

## 2.6 データベースの処理に関わる総コスト

以上の式を総合して、データベースをアクセスする際に必要となる全コスト  $C$  は、

$$C = (\mu_1 C_1 + \mu_2 C_2 + \mu_4 C_4) \quad (2.22)$$

条件

$$\sum_{k=1}^P \sum_{j=1}^S x_{ijk} = 1 \quad (i=1, 2, \dots, A)$$

$$n_{kj} \leq m_{ckj} \quad (k=1, 2, \dots, P; \quad j=1, 2, \dots, S)$$

$$x_{ijk} = 1 \text{ or } 0 \quad (k=1, 2, \dots, P; \quad j=1, 2, \dots, S; \quad i=1, 2, \dots, A)$$

で計算するものとして、次節のシミュレーション実験に適用する。

### 3. シミュレーション実験

前節で定義した処理コストの定義式に従いシミュレーション実験を行う。本研究では、2 拠点、3 拠点に分散化した場合について実験を行った。分散化を行った場合にも、その下でさらにファイルの分割を考慮することが当然重要だと考えられるので、我々は、この点も考慮してファイル分散を考えた。しかし、分散と分割の両方を同時に考慮すると、ファイル分散配置の組み合わせ数が膨大になるので、ここでは、それらのうち代表的と思われるいくつかの場合についてシミュレーション実験を行った。もちろん、本シミュレーションで用いた分散・分割方法が最良であるかどうかについては議論の余地があるが、少なくとも全体の傾向を把握するためには十分であると考ええる。2 拠点、3 拠点に分散化した場合のデータベース設計の特徴については表2、表3に略記した通りである。

分割法	分散・分割の特徴
1	一方にデータベースを設置し、他方はそれを利用する
2	データベースを両拠点に分散、関係の強い情報を集める
3	一方のデータベースのみを2分割
4	両方のデータベースを2分割
5	両方のデータベースで、各々無作意に2属性ずつに分割
6	両方のデータベースで、各々1属性ずつに分割

表2 2 拠点へのファイル分散の特徴



分割法	分散・分割の特徴
1	1 拠点にデータベースを設置し、他はそれを利用する
2	2 拠点にデータベースを分散し、残り 1 拠点はそれを利用する
3	3 拠点にデータベースを分散
4	3 において、各拠点でデータベースを 2 分割
5	各拠点で、各々無作意に 2 属性ずつに分割
6	各拠点で、各々 1 属性ずつに分割

表3 3 拠点へのファイル分散の特徴

シミュレーション実験は、つぎの 5 段階のステップに分割して行った。

(1)第 1 段階

あらかじめ設定しておいた分散・分割法に対応したパラメータをセットするとともに、各拠点間の転送コストの比率を設定する。

(2)第 2 段階

前段階で与えられた分割方法に基づいて、データベースの格納コストを計算する。

(3)第 3 段階

計算機内で一様乱数を発生させて、表 4 に与えられたような処理形態別の要求発生頻度に基づいて、処理形態と参照拠点及び相手先拠点を決定する。

(4)第 4 段階

第 3 段階で決定した処理形態によって、処理コストを計算する。

(5)第 5 段階

第 2・4 段階で得られた処理コスト等の統計量を収集し、出力する。

本シミュレーションにおいて、乱数は一様乱数を使用した。それは、データベースのアクセス要求は、システム全体の勤務時間帯を通して一定であり特定の時間帯には集中しないと考えられるので、指数乱数や正規乱数のような特定の出現区間に集中するような乱数は使用しなかった。また、転送コストについては、同一拠点内での転送コストと

属性	処理要求 関係拠点 処理頻度	読み込み												変 更			追加・削除
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
		1	1	2	2	3	3	1,2	2,3	3,1	1,2,3	1,3	2,3	1,2	2,3	3,1	1,2,3
属性	処理頻度	4	2	5	9	2	12	4	2	52	12	52	1	52	12	4	12
1		1.	1.					.3			1			1			1
2		1.	.42					1.			1		.3	1			1
3				.7					.8		1		1.		1		1
4				1.	.9			.5			1			1	1		1
5						.4				.9	1		.7	1		1	1
6						.4	.2		.8		1				1	1	1
7					1.				1.		1				1		1
8				.7				.2			1		.8				1
9						.4				.3	1		.6	1			1
10							1.		.7		1		.6		1		1
11						.4				.8	1		.4				1
12						.4			.6		1		.5				1
13					.9			.8			1						1
14							.2			.8	1		.2				1
15		.5							.7		1		.9				1
16			.42					.3			1						1
17				.9							1		.5				1
18				.7				.3			1		.2		1		1
19			.42						.6		1			1			1
20						.2		.5	1.		1						1
21						1.	.2	.4			1				1	1	1
22		.5								.6	1	1.				1	1

表4 処理別発生頻度と属性の使用確率

他拠点間の転送コストの比率を1 : 1, 1 : 5, 1 : 10, 1 : 50, 1 : 100の5種類に変化させた場合について比較検討した。

#### 4. シミュレーションの結果

図1は、2拠点に分散化した場合の同一拠点内の転送コストと他の拠点からの転送コストを1 : 1としてシミュレーションを行った場合の結果である。一般に、ブロック長が大きいときには分散後の分割数が小さい方がコストが少なく、ブロック長が小さいときは分割数が大きい方がコストが少なくなる傾向があることがわかる。ブロック長を大きくしてしかも分割数を増やすと、格納コストは増大するが処理コストは減少する傾向にあり、両者を重ね合わせると結果的に“U”字形になって現れるのである。ブロック長を小さくしたときは、分割数を増加しても格納コストの比率がほとんど増加せず処理コストの方がはるかに大きいため重ね合わせると図示したように右下がりになるものと思われる。

つぎに、転送コストの比率を同一拠点間と他の拠点間で1 : 1, 1 : 5, 1 : 10, 1 : 50, 1 : 100と変化させた場合の総コストへの効果について中央集中型と比較した場合について調査する。同一ブロック長の場合には、転送コストの比率が大きい方が相対的成本は減少している(図2)。一方、各分割法において、転送コストの比率を1 : 1とした場合と1 : nとした場合を比較すると、転送コストの比率が小さいときは分割数を増やした効果は小さいが、コスト比の差が大きくなると分割の効果が歴然と現れる(図

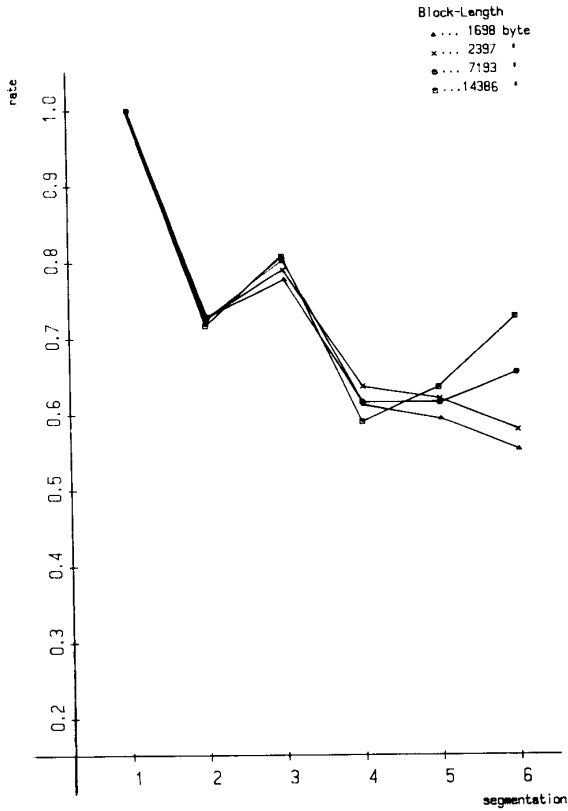


図1 2分散の場合の総処理コストの変化

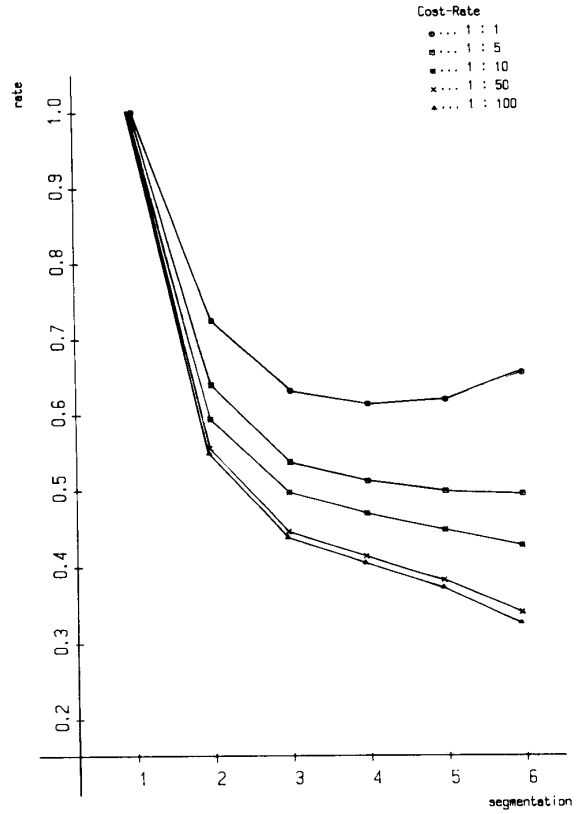


図2 2分散：転送コストによる総処理コストの変化（中央集中型を基準）

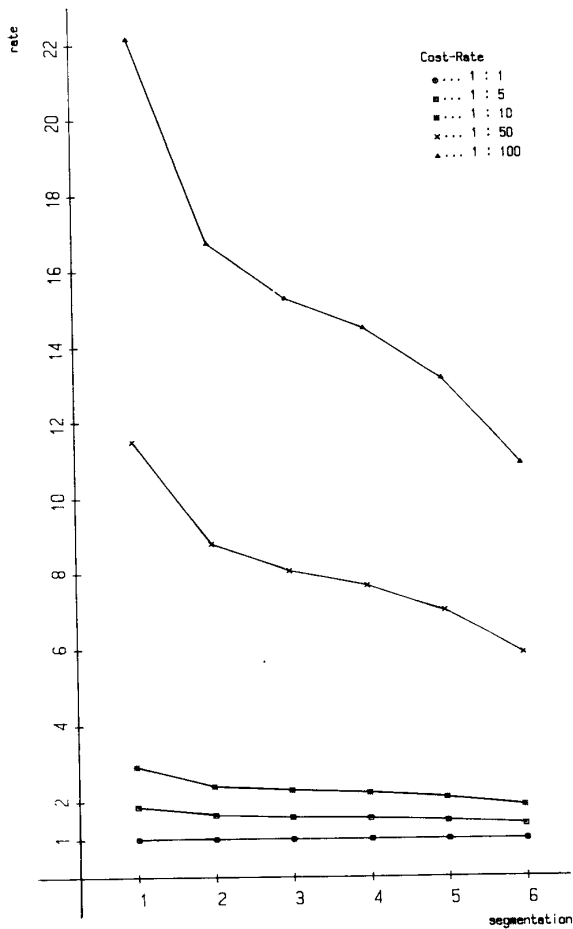


図3 2分散：転送コストによる総処理コストの変化（各分割法で比率1を基準）

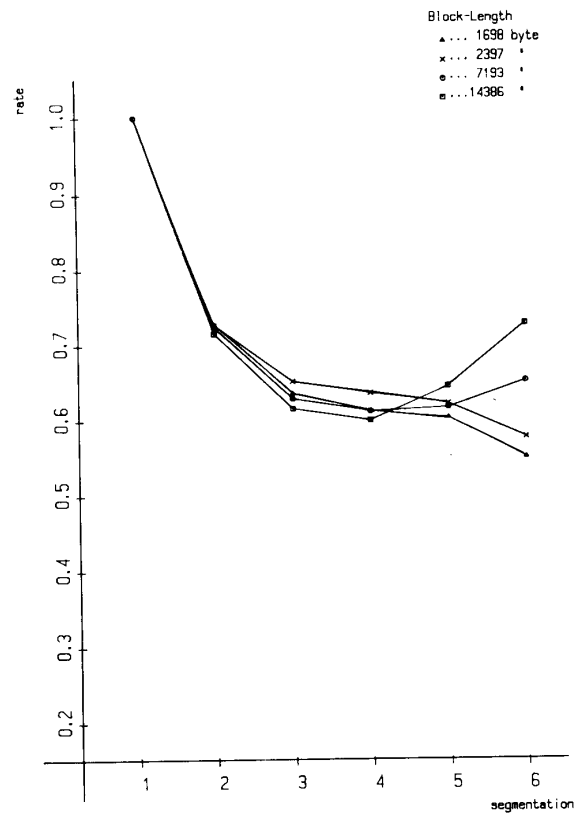


図4 3分散の場合の総処理コストの変化

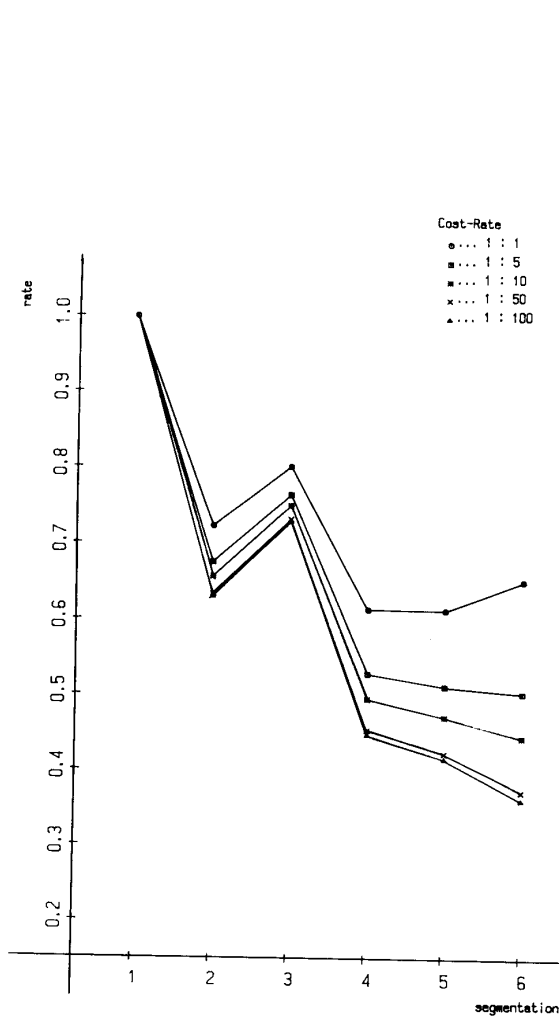


図5 3分散：転送コストによる総処理コストの変化（中央集中型を基準）

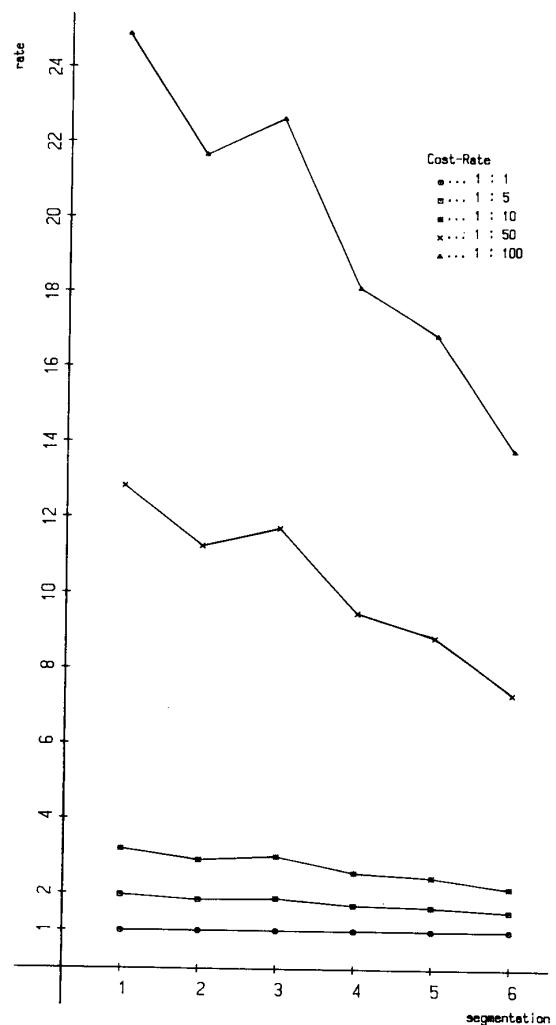


図6 3分散：転送コストによる総処理コストの変化（各分割法で比率1を基準）

3)。3 拠点に分割した場合も、同様な傾向が現れている（図4～6）。ここで、分散後の分割法3においてグラフの傾向が乱れているが、これは本シミュレーション実験に用いたデータの特異性に問題があるものと思われる。

## 5. 考 察

データベースの分散問題において、情報を各拠点に分散した後に各拠点で極端に分割するのは余り得策でないことが分かった。しかし、拠点間の転送コストの比率が高いときには分散後の分割数を増加させることが総処理コストを抑制する有効な手段であるといえる。これらの結果は、データベースを設計する場合の重要な指針となるであろう。

一般に、データベースの構築方法や利用方法はそれぞれのユーザーについて異なっていると考えられる。また、データベースに含まれるデータの属性や数、長さ、利用形態は異なり、分散すべき拠点数や分割すべきファイル数も当然異なっている。

多数の拠点で発生した大量の情報を効率よく低価格で処理できるようにするにはどのようにすればよいかという観点から情報の蓄積技法を眺めたとき、中央集中型より分散型のデータベースを構築した方が有利であることが実験から確かめられた。

つぎに、データベースの分散を考える場合の重要な鍵として、(2.22)で定義した総処理コストの計算式における各コストをどのようにして見積るかという問題が残っている。このことは、各処理システムの状況によって異なっている。例えば、磁気ディスクの格納コストはディスクの容量や使用効率によって左右されるであろう。そのような場合、ディスクのレンタル料あるいは買い取り料からバイト当りのコストを計算するのも一法であろう。拠点間の転送コストは、NTTの回線使用料から計算するのも良いであろう。また、ディスクのヘッド移動コストや、メモリ間との転送コストは確定した算出方法はないが、CPUの処理スピードや、データ転送スピード、データベースの使用頻度等から導き出すのがよいと考えられる。

分散型データベースを導入する場合、属性の利用頻度や処理頻度、利用確率、利用可能なリソースの量等にマッチした形で分散の度合を決定しなければならないことは言うまでもない。本研究結果によると、上語各種の環境条件に適合した分散を考慮したデータベースの作成が望ましいといえよう。その設計に当たっては、本研究で提案したコスト定義式とシミュレーション実験は十分応用できるものと考えられる。

## 参考文献

- [1] 青江, 成久: ファイルの有効なセグメンテーションについて, 岡山理科大学紀要, Vol.22-A, pp. 209-221, Mar., 1987.
- [2] J. A. Hoffer, "An integer programming formulation of computer data base design problems," Information Science, Vol. 11, pp.29-48, July, 1976.
- [3] 国友, 久保: データベース, 日本コンピュータ協会, 1983.
- [4] 東田: 属性モデルを用いたデータベースシステム, 情報処理, Vol.24, No.11, pp.1327-1335, Nov., 1983.
- [5] M. J. Eisner and D.G. Severance: "Mathematical techniques for efficient record segmentation in large shareg databases,," Journal of the Associatio for Computing Machinery, Vol. 23, No.4, pp.619-635, Oct., 1976.
- [6] J. M. Babad: "A record and file partitioning model," Communications of the ACM, Vol. 20, No.1, pp.22-31, Jan., 1977.
- [7] 高平, 鈴木: 分散型データベース技術, 情報処理, Vol.23, No.10, pp.931-938, Oct., 1982.

- 〔8〕穂鷹，若林：データベース管理，日本コンピュータ協会，1982.
- 〔9〕斉藤：データベース管理システム，日本コンピュータ協会，1979.

## **Studies on Planning for Effective Distributed Data Base**

Hiroyuki NARIHISA\* and Toshio AOE\*\*

\*Department of Electronic Engineering,  
Okayama University of Science  
Ridaicho 1-1, Okayama 700, JAPAN

\*\*Department of Early Education,  
Junsei Junior College  
Ige-machi 8, Takahashi 716, JAPAN

(Received September 30, 1987)

Today, we can easily use a various kinds of computers to process a large scale informations. Such large scale data are generated arround us and these data are collected to our computer systems to process them. It is important problems that how to organise these data.

In order to manipulate these large scale data, we introduce a data base consept, especialy for a distributed date base concept. A distributed data base system is any data base system involving multiple sites connected together with some kind of communications network, in witch a user at any site can access data base storerd at any site.

In this paper, we represent an effective distributed data base on the computer network.